# Enhancing Cybersecurity Education through High-Fidelity Live Exercises (HiFLiX)

Johan Sigholm
Sloan School, MIT &
Swedish Defence University
johan.sigholm@fhs.se

Gregory Falco
CSAIL, MIT &
FSI, Stanford University
gfalco@mit.edu

Arun Viswanathan
Jet Propulsion Laboratory,
Caltech
aviswant@jpl.nasa.gov

## Abstract

*The people responsible for building the IT products and infrastructure of tomorrow – today's students of the computing disciplines – oftentimes do not have the opportunity or proper motivation to develop cybersecurity skills meeting the needs of the job market. This paper introduces High Fidelity Live eXercises (HiFLiX) a teaching/learning activity designed to expose students to cybersecurity challenges resembling those they could face in a future work environment. We describe a HiFLiX prototype study, conducted as a collaboration between the Massachusetts Institute of Technology's CyberSecurity @CSAIL research group and NASA's Jet Propulsion Laboratory. Our analysis indicates that the proposed delivery method met the stipulated cybersecurity educational outcomes and increased the motivation for future cybersecurity studies in the majority of participants. Two previously unknown software flaws were also discovered.*

## 1. Introduction

The rapid spread and adoption of Information and Communications Technology have put computing[1] engineers in high demand. However, as products, services and even critical infrastructures have become targets of antagonistic cyberattacks, recruiting computing experts who also have an adequate understanding of cybersecurity[2] has quickly become a priority. Nevertheless, recruiting talent in this field has proved to be a challenge. Hiring managers are struggling to find candidates for open positions and estimates project that several million cybersecurity-related positions will be vacant in the coming years [1][2].

While the number of Bachelor's and Master's graduates from engineering programs within the computing sciences have doubled over the last decade [4], students are generally not provided with the learning options leading to the cybersecurity expertise sought after by the market. Several reports have pointed to the need of developing curricular guidelines for cybersecurity education [5][6].

The relatively few students who have studied security at university have oftentimes been exposed to courses that are entirely theoretical, dealing with principles and concepts rather than practice [7]. While a class on computer security may cover such foundational principles as the Bell-LaPadula model [8] in depth, the students seldom get the chance to apply their knowledge in practice, to develop independent thinking, or to experience what a real-world cybersecurity challenge may entail. While it is important that students learn the theory (the "why"), training them to apply theory (the "how") is also needed [9]. As recently noted by the ABET accreditation organization, the incorporation of hands-on and practical cybersecurity instruction is a key part in modernizing engineering education [10].

In this paper, the research question we seek to answer is how practical cybersecurity exercises could be designed to contribute significantly to educational outcomes in this area. Our proposed delivery method targets the use of hands-on cybersecurity skills. This is done by exposing students to real-world systems and mentoring them through realistic challenges, through a collaboration between academia and industry.

The paper is structured as follows. Section 2 reviews previous literature, Section 3 describes security training exercises, Section 4 describes the experimental setup, Section 5 outlines the results, which are further discussed in Section 6. Our conclusions are offered in in Section 7.

## 2. Related work

A considerable number of research papers published during the last four decades deal with investigations towards software engineering education

[1] Refers to the five ACM computing disciplines Software Engineering, Computer Engineering, Computer Science, Information Systems, and Information Technology.
[2] Defined as a "computing-based discipline involving technology, people, information, and processes to enable assured operations. It involves the creation, operation, analysis, and testing of secure computer systems." [3]

URI: https://hdl.handle.net/10125/60193
ISBN: 978-0-9981331-2-6

from many different points of view. This section provides an overview of the major trends in cybersecurity education during the last decades and suggests some relevant literature.

Although the nomenclature has evolved, the question of how to best teach engineering students cybersecurity is far from new. Many of the founding principles of cybersecurity, at least in a technical sense, can be traced back to the 1970s. The seminal paper "Security Controls for Computer Systems" [11] broadened the scope of computer security to include protection of the data itself, limiting unauthorized access. The "CIA triad" of confidentiality, integrity and availability defined in 1975 is a core tenant of cybersecurity education [12].

By way of the World Wide Web, the Internet was brought to wider use during the end of the 80s. This increased the importance of cybersecurity, as emphasized by the ACM Task Force on the Core of Computer Science [13]. Similarly, in their Master of Software Engineering Curriculum from 1989, the Software Engineering Institute (SEI) at Carnegie Mellon University suggested that topics such as security, along with privacy and software piracy, should be discussed in context in all courses to set examples for students [14].

In the 1990s, security started to become more visible in various engineering curricula. For instance, in the then emerging field of real-time computing, security had a natural role [15]. At the 1996 IEEE Symposium on Security and Privacy, a special panel was held to gather input from industry on cybersecurity teaching requirements [16]. In their paper "Integrating Security into the Curriculum" [17] Irvine, Chin and Frincke establish that "the education [in engineering programs] must result in graduates prepared for the security challenges they will encounter in their professional roles," but that "too few computer science and engineering programs today [1998] pay adequate attention to security". They also underscore that educational outcomes must be consistent with those of the larger engineering context. This is in accordance with the established concept of Constructive Alignment, used within the subject of pedagogy in higher education [18].

Bertrand Meyer [19] highlights the importance of "teaching by doing," an approach necessary in preparing students for professional cybersecurity challenges. In 2004 a joint ACM-IEEE task force on Computing Curricula published the Software Engineering 2004 (SE2004) Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering [20]. While SE2004 includes several components relating to security, security still was not defined as an independent knowledge area.

The President's Critical Infrastructure Protection Board [21] illustrated in 2003 how flaws in computer software can lead to serious vulnerabilities in critical infrastructure. For this purpose, the U.S. Department of Homeland Security (DHS) formed a group with the task of defining a Common Body of Knowledge for secure software assurance [22], where one of its stated goals was to drive curriculum development in academic institutions. The Carnegie Mellon SEI was subsequently enlisted by the DHS to develop a curriculum for a Master of Software Assurance degree program [23]. In the paper "Foundations for Software Assurance" Woody, Mead and Shoemaker define principles underpinning a curriculum for such a Master's program [24]. Tom Hilburn and Dan Shoemaker [25] discuss the security competency required in the software engineering profession and present models for competency management.

In 2015 the Joint Task Force on Cybersecurity Education was chartered by the ACM Education Board. Their publication Cybersecurity Curricula 2017 [26] is the to-date latest progression in curricula development in the field. Important learning outcomes include forensic analysis, penetration testing, ethical hacking, and offensive techniques.

## 3. Security Training Exercises

Hands-on computing exercises have been demonstrated as a very effective method of teaching and learning cybersecurity [27]. Such exercises engage learners and allow them to practice and hone essential cyber-skills, recognized as a necessity by the majority of the educational and professional communities [28]. There are several methods for introducing students to security problems using training exercises. Perhaps the most well-known are Capture the Flag (CTF) competitions. CTFs are designed to encourage students to find vulnerabilities in a designated computing environment that was built for the specific purpose of training. The predetermined vulnerabilities are generally marked by "flags" which consist of a string. Upon discovering the vulnerability, CTF participants "capture" the flag by copying and pasting the string into a validation website to receive points.

There are two types of CTFs – Jeopardy and Attack-Defense. For Jeopardy CTFs, participants are provided categories of security issues and techniques that must be found and executed (e.g. Web, Binary, Reversing.) In many cases, the categories are sequential. For Attack-Defense CTFs, all teams are provided computing infrastructure they must defend while simultaneously tasked with attacking others' infrastructure. Flags are recovered when identifying

and submitting vulnerabilities in the team's own infrastructure which are subsequently patched. Flags are also recovered when a vulnerability is found in an opposing team's infrastructure. Points are awarded for submitting these flags. CTF rules of engagement, and point distribution vary across CTFs. For examples of different CTFs, see [29][30][31].

While CTFs provide a dynamic environment for students to learn about security, there are several disadvantages to their structure. First, CTFs are high intensity environments that require participants to have a knowledge base of security to be successful. There is very little opportunity to teach new skills during a CTF. Therefore, it is easy for less experienced participants to get left behind those that have more experience. Participants can of course learn by watching, but it is not an optimal environment for learning new skills due to the time pressure. Another learning challenge during Jeopardy style CTFs is that they are structured in category stages where capturing one group of flags for a category – such as Web, will unlock the next category of flags. The sequential unlocking of tasks limits participant's educational exploration because it is easy to get stuck on a certain task and be unable to try different exercises.

On a higher level, CTFs also lack a fundamental component of cybersecurity education – helping students connect the dots between existing vulnerabilities and their origin. When breaking and entering, and quick-and-dirty patching, is the rewarded driver of an exercise, focus on understanding cybersecurity principles and their best practice application is lost.

One of our goals in developing the High-Fidelity Live Exercises (HiFLiX) was to encourage participants to think in terms of the "cyber kill chain" – a description of the sequence of steps often used by adversaries in real-world situations [32]. Rather than to follow sequential predefined tasks we wanted the students to consider how the model could be used to attack a system, but simultaneously what flaws that allowed them to do so, and what security controls that could be applied to prevent a successful attack. Further, we also wanted to capture participant attack data which would mimic real adversarial activity.

To achieve realism in both exercise environment and attack patterns, we designed a system with networked hosts running NASA mission software and services, including any pre-existing vulnerabilities. The exercise would thus simulate a real-world environment with high fidelity. Our hypothesis was that the participants would find it more educational and motivating to be exposed to real-world systems and challenges, as opposed to the limited framework-style environments that students are commonly presented with in classroom labs.

# 4. Experiment

## 4.1 Overview

As mentioned in the previous section, our proposed HiFLiX differs from other forms of competition-style exercises on several accounts. It introduces the participants to a real-world system environment, a replica of a system used in production, instead of a fictitious, competition-specific system. It also runs over a longer period of time (in this case five days) and offers continuous instructional support and guidance. The exercise is not set up as a game or competition, but rather a learning experience with a partially open-ended set of objectives. The experimental HiFLiX described here was arranged as a collaboration between the Cyber Defense Engineering and Research Group at the NASA Jet Propulsion Laboratory (JPL) and the Massachusetts Institute of Technology's (MIT) CyberSecurity@CSAIL research group. It was developed by JPL and offered to students at the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) during the last week of MIT's four-week winter break independent activities period (IAP) in January 2018. As the experiment was hosted virtually using cloud services, the students pre-registered for the experiment could take part either from a computer lab at MIT, from their own home, or any other place they chose.

The experiment had the following main goals:

(i) Evaluate if this type of exercise would satisfy cybersecurity learning outcomes;[3]
(ii) Gather data for cybersecurity-related research and development tasks; and
(iii) Improve awareness of new attacker tools, tactics and procedures.

A total of thirteen students actively participated in the experiment.[4] They were tasked with achieving a set of HiFLiX objectives in a network environment configured to closely resemble a NASA mission system. These included identifying and documenting system weaknesses that could be exploited to extract information. Some well-known exploitable flaws and misconfigurations were intentionally planted, while

[3] Targeting the *understanding* and *applying* levels of the Bloom's Revised Taxonomy, see: http://ccecc.acm.org/assessment/blooms/.
[4] Only about half of the students who had initially signed up to participate ended up doing so.

also allowing for the possibility of finding previously unknown vulnerabilities. Six students completed at least one of the stipulated objectives, and two students completed all the objectives. In addition, two previously unknown Cross-Site Scripting (XSS) vulnerabilities were found in mission software.

## 4.2 Design

We summarize our design choices and rationale into four categories, described below.

**Problem design.** The HiFLiX was organized to encourage participation across a mix of hacker skills.[5] A pre-event survey indicated that almost 52% of the potential participants were *newbies* with very little or almost no knowledge of hacking, 26% were *familiar* with hacking and 21% were *experts* in hacking. This mix of skillset meant that our HiFLiX design had to contain a mix of simple and complex problems to be accessible and to be of interest to all types of participants, from *newbies* to *experts*. Our problem design strategy involved the following.

Traditionally, security exercises are very time constrained. This limits the ability for students to learn at their own pace. Further, objectives for security exercises that are staged as described above limits students' ability to try new objectives when they are stuck on a problem. To avoid these constraints, we attempted to apply a constructivist learning model to the HiFLiX. This Montessori-style pedagogy is intended to promote discovery and diffuse frustration caused by traditional educational training boundaries. The Montessori pedagogy has been shown to be successful beyond primary education [33]. In the spirit of this educational approach, we chose to fully virtualize the HiFLiX environment, maximizing the flexibility of the exercise. We provided a window of five days to complete the HiFLiX so the students can learn and approach the objectives at a speed with which they are comfortable. Also, we chose to allow students to attempt any objective in any order. This design decision was to enable participants to start with whatever objective they found most interesting or accessible so that they did not become discouraged by the exercise.

We chose to rely on usage of such well-known and freely available cybersecurity tools as Metasploit[6] and John the Ripper[7] to solve most of the HiFLiX problems. To make it easier, we provided students with the popular penetration tester's toolkit Kali Linux,[8] a popular Linux distribution which comes pre-packaged with a variety of tools for digital forensics and vulnerability identification. This ensured that the *newbies* would get a chance to learn new yet very relevant tools by participating in the HiFLiX, while the *familiar* and *expert* category participants could hone their skills in the usage of these tools, or use their own if they so preferred.

We embedded a few well-known vulnerabilities and very common system misconfigurations to ensure that all students could make progress in the HiFLiX and advance their cybersecurity skills by quickly identifying exploitable vulnerabilities common in real software and systems. For example, we planted the well-known Bash Shellshock vulnerability (CVE-2014-6271)[9] and the more recent SambaCry (CVE-2017-7494) in a few of the systems. These vulnerabilities could be discovered directly through Metasploit.

We planted some advanced vulnerabilities which required some research and thought to challenge the *familiar* and *expert* category participants. For example, we planted a Java deserialization vulnerability in one of the systems, which required participants to first discover this vulnerability, and then use a combination of tools to verify that it could be exploited.

Finally, we provided the *experts* with a system containing real mission software and services and challenged them to find previously unknown (zero-day) vulnerabilities. Overall, we designed at least two of the three HiFLiX objectives (described in Section 4.5) to be solvable by most participants and had one objective specifically targeted at the experts.

**Infrastructure design.** Our original strategy for the HiFLiX consisted of us sending pre-packaged virtual machines to each of the individual participants. This was an easy approach and would have eliminated any requirement for complex infrastructure setup and management, but unfortunately it also made it impossible for us to gain any real time visibility into the actions of the participants, and monitor data from the hosts in any meaningful way. It would have also made it difficult for us to troubleshoot problems and assist the participants. Remotely hosting the HiFLiX on JPL's infrastructure was another option but it was fraught with security concerns and addressing those concerns would have required a massive effort both in terms of procuring the necessary approvals and the engineering. We finally settled on hosting the HiFLiX remotely on the Amazon AWS cloud. This enabled us

[5] Here referring to aptitude in applied cybersecurity, including computer system vulnerability evaluation and penetration testing.
[6] https://www.metasploit.com/
[7] https://www.openwall.com/john/
[8] https://www.kali.org/
[9] https://cve.mitre.org/

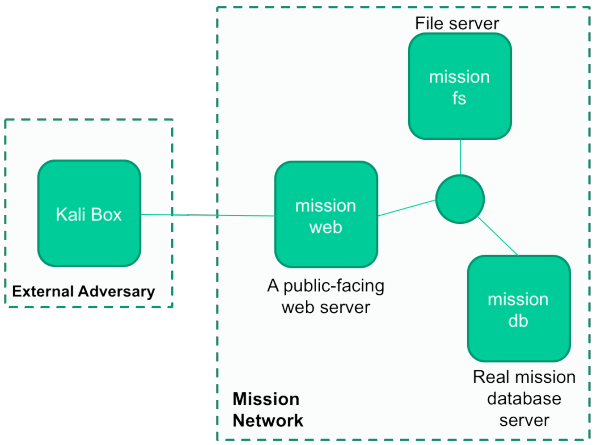


**Figure 1. Simulation of a portion of a real mission network**

to construct a sophisticated and secure virtual environment for each participant within a couple of weeks. In addition, this also allowed us to fully instrument the virtual machines and the virtual networks for monitoring purposes and provided us near complete control over all the aspects of the environment, utilizing a web dashboard from the West coast with the students located on the East coast. Hosting the HiFLiX on the cloud also allowed us to easily scale to a large number of participants within a matter of minutes. In the end, this proved to be a cost-effective solution both in terms of the time required to setup and manage, and the overall cost of the resources required to make the event a success.

## 4.3 Technical environment

We simulated a portion of a real mission network for each participant, with three virtual machine hosts. The *missionweb* simulates a web server which is accessible from the internet. We use a stock install of Red Hat Linux 7.3 along with a version the Apache web server. The *missionfs* simulates a file server, which is accessible only from the web server and other machines inside the network. We again use a stock install of Red Hat Linux 7.3 with various file sharing software such as ftpd, NFS and smbd installed on it. The *missiondb* is a real mission database server and contains the base installation of real mission software. All these three hosts are grouped into a single subnet. The hosts were also logically isolated by way of port and protocol filtering so we could control what was visible from one host to the next. This provided us the ability to slowly expose more to the attacker as they progressed through the network from host to host.

Another subnet was used to simulate the external adversary and consisted of one virtual machine containing the Kali Linux distribution. Students were only given direct access to this Kali Linux host, and they had then to navigate their way into the mission network to accomplish the HiFLiX objectives.

The environment, shown in Figure 1, was replicated for each participant in the AWS cloud. Each environment was instantiated as an Amazon Virtual Private Cloud (VPC), which provided logical isolation between each participant and ensured that they could not interfere with each other's environments.

Each host was instrumented to collect exercise data, including network traffic, shell command history, disk transactions, and system logs. In addition, flow logs were enabled for each VPC in the AWS cloud environment. This enabled monitoring the traffic coming into and out of each VPC and between the subnets. All data, except packet captures, was fed into a Splunk[10] infrastructure in real-time, and each participant's data was tagged uniquely for future attack analysis.

## 4.4 Vulnerabilities and misconfigurations

The following is a summary of the vulnerabilities and misconfigurations planted across the hosts.

**missionweb** contained the Heartbleed (CVE-2014-0160) and Shellshock vulnerabilities exposed via the Apache webserver, in addition to an SSH misconfiguration which contained weak and easy brute-forced passwords for several accounts on the system.

**missionfs** contained vulnerabilities in the proftpd FTP server, dnsmasq service, the sambacry vulnerability in smbd, a privilege escalation vulnerability in the sudo binary, and an NFS misconfiguration which allowed the whole directory structure to be mounted as root.

**missiondb** contained real mission software and was largely left untouched to encourage discovery of zero-day vulnerabilities. The only vulnerability planted was a Java deserialization vulnerability which was added on top of the existing vulnerable ActiveMQ software running on the host.

## 4.5 Objectives and Rules of Engagement

A welcome email was sent to the participants a couple of days before the exercise. It described the overall HiFLiX infrastructure, laid out the objectives, set out the rules of engagement and the reporting requirements.

The participants were made aware that the HiFLiX exercise was designed to accommodate different levels of previous experience in working with cybersecurity,

[10] https://www.splunk.com/

and cater to a mix of hacker skills. For beginners, a main benefit would be to familiarize themselves with new security tools to discover and document realistic vulnerabilities in systems configured to resemble those supporting NASA's missions. Students with more experience in penetration testing would have an opportunity to explore complex attack scenarios that may require use of advanced exploitation techniques to discover unknown vulnerabilities. The three objectives that were set forth were the following.

*Objective A* was to penetrate the simulated mission network and extract information from the mission telemetry database (identified as missiondb). The specific information to extract consists of an administrative username and password stored within the MySQL database running on that system.

*Objective B* was to find and report at least eight of the vulnerabilities (either planted or zero-days) along with a proof-of-concept across the systems.

*Objective C* was to perform a security assessment of the mission telemetry database and find at least one significant configuration error or vulnerability in that system.

A Slack channel (a popular messaging platform) was setup for communication between the organizers and the participants, and between the participants themselves. Students were encouraged to offer general support to other participants when possible but to refrain from providing specifics or clear paths to the solutions. Tips and hints regarding the exercise were shared over Slack as the competition progressed. Participants were also made aware that log and packet data from the event was being collected. A list of rules of engagement was shared with all the participants. This included bans on disabling any of the exercise monitoring tools, transferring content out of the exercise environment, disclosing any discovered vulnerabilities, and sharing used exploits.

## 5. Results

The experiment ran according to plan and was successful in terms of student participation, how well they were able to fulfil the exercise objectives, and the amount of data gathered from the HiFLiX. In total, thirteen students participated in the experiment, with six students completing at least one objective. Two students completed all the objectives.

### 5.1 Participant survey

Eight of the thirteen participants answered the post-event online survey, consisting of a set of questions with multiple choice as well as free text answer options. While we acknowledge that the sample size is not statistically significant and thus does not allow for declarative judgement, the results are used as qualitative indicators of success. The respondents were asked to answer questions related to their personal impression of the exercise, to rate its perceived difficulty level, to comment on its design and setup, to describe their own cybersecurity competence level before and after the exercise, and to estimate of how well they believe they had reached the educational outcomes of the exercise. Some of the more interesting results of the survey are brought up below.

Six respondents were fully satisfied with the exercise and none were disappointed. Two respondents felt that the exercise could have been improved, mainly referring to technical issues related to reaching the objectives and to a perceived lack of documentation.

Five respondents found the exercise at the right difficulty level, whereas three thought that is was too hard. Six respondents found the allotted time for the experiment (five days) to be adequate, whereas two found it too short. Six respondents would prefer this type of exercise over a traditional Capture-the-Flag competition. Several respondents mention that being exposed to real software and vulnerabilities increased their motivation for the exercise. All the respondents answered that they had learned a lot or gained experience they considered to be valuable.

In the free text answers, several interesting comments were given by the respondents. One person mentions that a great benefit of this exercise, in comparison to previous CTF experiences, was its educational purpose. "This [exercise] helped us learn a lot since the staff was on standby to help guide, or give hints to, the students." Another respondent also mentions the "responsiveness" of the staff to assist and instruct during the exercise. Several respondents mention appreciating working on "real mission servers," and indicate that this was what had motivated them to take part in the experiment. "I enjoyed having to start with nothing and check everything for possible vulnerabilities, along with how to exploit them."

Of the more critical feedback provided in the survey, most comments were focused on challenges with the exercise. "I was confused about what I should do in the first place, but that could be because I'm inexperienced in [finding vulnerabilities in] real software." One respondent mentions spending too much time on a task which did not contribute to reaching the objectives, but goes on to reflect on that this also resulted in a learning experience. Suggestions given for future improvements include handing out instructions longer in advance of the exercise and providing more hints to the participants during the exercise days.

## 6. Discussion

As this study is limited in scope, and the number of participants in the experiment was relatively small, we cannot claim that the outcomes are replicable and generalizable for future exercises. However, we believe there are considerable insights we can draw from this research, to continue improving methods for hands-on cybersecurity education.

**Choose-your-own-adventure works.** In scoping the HiFLiX, we aimed to develop a learning experience tailored to the individual student. This entailed meeting the students at their existing level of cybersecurity education. In the spirit of Montessori approaches, and unlike most CTFs, there were no qualifiers, minimum skill requirements/prerequisites. This enabled students who were complete novices to explore a new software environment and promoted their general software engineering exposure by allowing them to explore a JPL mission system. Some participants commented it was their first time using Kali Linux tools and the ability to test the tools out in a safe place was valuable. The more experienced participants were motivated to find zero-day vulnerabilities in the environment because it was a real mission system. When problem sets are designed for CTFs, experienced participants get into the habit of searching for certain vulnerabilities that are token problems for such exercises. Because the software engineering team that developed this real mission system did not even know all the vulnerabilities therein, the students were encouraged to think outside of the normal CTF-type flags. This was reflected in one student's feedback on "start with nothing and check everything." Additionally, a motivation for these security exercises is always bragging rights. Finding zero-days for a NASA JPL mission system provided ultimate bragging rights – especially for MIT hackers. Nevertheless, partnering with an industry or government actor from the local area/region would in essence fill the same educational purpose.

**Support is essential.** Student feedback revealed that the choose-your-own-adventure was only successful because of the support provided by the JPL team. The structure of traditional cybersecurity exercises provides boundaries and comfortable tasks to students. The comparatively less rigid environment of the HiFLiX required a support infrastructure that enabled quick response times to student queries. Part of meeting the students where their skills were required hands-on educational communication. Students often asked the support team questions like "how do I use a XYZ tool" or "what can I try next?" The success of this HiFLiX relied on bringing a live classroom experience to a live security exercise, where fundamental security tests and approaches were taught virtually while a student tried to "hack away" at the exercise. While some CTFs may have labeled this hands-on teaching environment as "cheating," we encouraged it so students would feel supported in their experience and not abandon the exercise in frustration.

**A win-win-win for academia, industry and students.** Developing security exercises are generally either a labor of love or a source of revenue. Academic institutions build security exercises for students to inspire future generations of security researchers and analysts. Organizing such events is no easy or inexpensive feat for the academy, considering the time and resources required to develop problems, transport students to the cyber range and host the infrastructure. Beyond academic institutions hosting security exercises there is also a cottage industry of security companies hosting events targeted at hackers training for high-profile security competitions such as DEFCON. Hackers pay to participate in these events and trainings. In this case, MIT's CyberSecurity @CSAIL research consortium partnered with NASA JPL to host the event virtually. Through this partnership, MIT was able to provide cost-free security education to its students. The costs were instead incurred by NASA JPL who spent considerable time designing and constructing the exercise, as well as supporting the participants.

Nevertheless, JPL realized considerable value from hosting the event. An ongoing challenge for JPL, as for many other organizations, is attracting high quality software engineering security talent. This exercise provided a forum for JPL to connect and work with talented MIT students who expressed interest in learning more about their work, thus paving the way for recruiting opportunities. Compared to traditional recruiting methods, JPL was able to interact with these students and determine who might be a fit for their organization based on how the participants approached the exercise technically and even socially.

Another major benefit to JPL was that the relatively open-ended forum allowed the participants to search for zero-day vulnerabilities against the mission system. Participants' attacks on the mission system during the HiFLiX were captured for analysis. JPL is now using this data to identify common attack patterns against their mission systems to better detect attacks as they occur against their networks. Two zero-day vulnerabilities were found by the students which JPL was able to verify, mitigate and ultimately patch. As a result, the HiFLiX acted as a low-cost security audit for the mission system.

From a student perspective, the benefits of participating in the exercise are perhaps obvious, but worth reiterating. Students were able to learn security tools and techniques in a safe environment without the constraints of CTFs and other more traditional security training events. Further, students were working on a real-life mission system that was representative of security challenges that are found in industry. Finally, students received real-time support from the teams that work with these mission systems daily and were given guidance as to how to approach the objectives presented.

**A remote virtualized environment is an enabler.** When the HiFLiX was in the early stages of design, we considered flying MIT students out to Pasadena, CA where NASA JPL is located, so they could run security exercises on local mission systems. Considering the transnational travel and security considerations due to the sensitive nature of systems at JPL, the event seemed financially and logistically challenging to execute. Virtualizing the mission system environment provided us the flexibility to accommodate all students interested in participating without concerning ourselves with the logistics of travel and garnering student credentials to the JPL site. Hosting this event in the cloud also enabled us to successfully conduct the HiFLiX exercise without directly exposing JPL networks or JPL system resources to the attacks carried out by the participants. The virtualized environment was also inexpensive to host in Amazon Web Services (AWS) where we stood up 80+ systems and 20 virtual private clouds for a total cost of $2,327.32. The whole setup, from conceptualization to implementation, took slightly over a month with about 200 man-hours of effort. AWS also provided tools for usage analytics and other data collection capabilities which were valuable after the event was completed, to support future improvements of the exercise. Future HiFLiXs should also consider conducting remote and virtualized environments to provide flexibility to students, increased security control to the industry partner that provides a real-life system for the exercise and to minimize expenses.

**Thoughts on HiFLiX replicability.** We acknowledge that this study was conducted in an environment and with a partnership primed for success. Not every institution has the academic resources of MIT or the intellectual resources of NASA. To successfully replicate the HiFLiX in other settings, we will share some lessons learned in the organizational setup of the exercise.

Some may argue that MIT is a unique institution because of its student base and academic resources, which could limit the broader applicability of HiFLiX. We would, however, argue that there are good reasons to assume that a similar approach could be successful also elsewhere. The students who took part in the exercise all had different levels of previous experience, and were at various points in their computer science education. There were no location-based benefits for these students (everything was conducted virtually) and the resources provided to them were all open-source and freely available. A similar spectrum of students at other schools could thus well have equivalent experiences. The guidance provided to the students by MIT CSAIL security researchers centered around tips on using the open-source tools. There is no doubt that other security researchers and instructors could provide similar general guidance.

While in the case of this HiFLiX, NASA JPL made a sophisticated mission system available to the participants, this need not be the case. We believe the value of the exercise was not in the sophistication of the system provided, but rather in the realness of the mission system. With this said, it is not crucial for the industry partner to create something new or intricate for the students. Any real system can provide great value compared to fabricated environments, and any strong partnership between an academic institution and industry could be the starting point for a successful HiFLiX. In other words, the intellectual resources of NASA are not required to replicate this learning experience.

MIT's CSAIL and NASA's JPL had been working together on various research and development projects for over a year before running the HiFLiX. This enabled the JPL team to learn about the students at MIT who would be interested in such an exercise and select a mission system that has features that could appeal to these students. For example, most MIT students that are interested in security focus on systems. Rather than JPL building out a HiFLiX based on their robust database structures, JPL provided a mission system for the exercise. The pre-existing working relationships between the two organizations also enabled a degree of expectation setting for the exercise. Expectation setting was important so that there was no disappointing outcome for either party. Both MIT and JPL were aligned on what each respective organization hoped to achieve. The relationship also made communication between MIT and JPL before and during the exercise straightforward. The JPL HiFLiX lead was in constant communication with the MIT lead at CSAIL concerning potential issues and progress as the week progressed. Again, communication was simple because of the existing trust and rapport between the two parties.

## 7. Conclusions

In this paper, we have proposed a new delivery method for cybersecurity education. Introductory cybersecurity related skills and learning outcomes are in the process of becoming curricular requirements in most accredited computing degree programs. Our main contribution is thus given by offering an approach to achieving these learning outcomes, through the proposed HiFLiX concept. Although our experiment does not provide us with conclusive results as to the efficiency of the exercise as a teaching/learning activity, we believe that the approach is promising and merits future study and development. The preliminary qualitative results show that the exercise was valuable to students as well as the hosting organizations.

The first goal of the experiment was to evaluate if a HiFLiX exercise would satisfy intended cybersecurity learning outcomes. We conclude that most students who participated in the exercise demonstrated assessable proficiency in core cybersecurity concepts, through achievement of learning outcomes on the *understand* and *apply* levels of the Bloom's Revised Taxonomy.

The second goal, to gather data for cybersecurity-related research and development tasks was reached by the generation of an exercise dataset, and the third goal was reached by discovering previously unknown vulnerabilities in the NASA JPL mission system.

One of the main purposes of academic education is to teach students how to synthesize information, principles, concepts, and other materials to be able to apply it to novel situations. While understanding fundamentals is unarguably of importance, cybersecurity expertise also requires an understanding of tools, tactics and procedures and how they may be applied. In this paper we have addressed the question of how to enhance cybersecurity learning and to provide students in computing sciences with an opportunity to develop cyber-skills. Our proposed solution, the HiFLiX teaching/learning activity, shows that this can be done in a way that offers several positive outcomes.

The suggested exercises contribute in preparing the participating students for cybersecurity challenges they may encounter in their future professional roles and increase their motivation for pursuing subsequent cybersecurity studies. The exercises also promote constructive alignment between overarching security educational goals and intended learning outcomes of specific software reliability and assurance courses, by fostering skills required for independent, innovative and critical thinking.

Future work includes follow-up studies of upcoming HiFLiX exercises during the spring semester of 2019. This includes a planned study on the potential benefit of incorporating them as a mandatory part of the curricula within an engineering program. We also plan to investigate how constructivist learning methods can be further used in HiFLiX exercises so they can become an even more accessible tool for security education, serving as a missing link between computer security education and cybersecurity training.

## 8. Acknowledgments

The experiment was setup and executed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Bryan Johnson and Eddie Babbe from the Cyber Defense Engineering and Research Group at JPL assisted with experiment setup and management. Matt Derenski from the JPL Office of the Chief Information Officer assisted in the creation of the cloud infrastructure.

The authors would also like to thank Howard Shrobe and Lori Glover of CyberSecurity@CSAIL for supporting the HiFLiX. The first author was supported by grants from the Swedish Armed Forces and the Fulbright Program, both which are gratefully acknowledged.

## 9. References

[1] ISACA, "State of Cybersecurity 2018: Workforce Development", Technical Report, Information Systems Audit and Control Association, 2018.

[2] Steve Morgan, "Demand for cybersecurity talent rises sharply", *CSO Online*, January 11, 2018.

[3] Diana Burley, Matt Bishop, Siddharth Kaza, David Gibson, Elizabeth Hawthorne, Scott Buck, "ACM Joint Task Force on Cybersecurity Education", in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, Seattle, WA, March 2017, pp. 683-684.

[4] B. L. Yoder, "Engineering by the numbers", American Society for Engineering Education, Washington, DC, 2017.

[5] Andrew McGettrick, Lilian N. Cassel, Melissa Dark, Elizabeth K. Hawthorne, and John Impagliazzo, "Toward Curricular Guidelines for Cybersecurity", in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, Atlanta, GA, 2013, pp. 81-82.

[6] Michael Hogan and Elaine Newton, "Report on Strategic U.S. Government Engagement in International Standardization to Achieve U.S. Objectives for Cybersecurity", Report 8074, Volume 1, National Institute of Standards and Technology (NIST), 2015.

[7] Matt Bishop, "What is Computer Security?", *IEEE Security & Privacy*, 1(1), 2003, pp. 67–69.

[8] David E. Bell and Leonard LaPadula, "Secure Computer Systems: Mathematical Foundations and Model", Technical Report M74-244, MITRE Corp., Bedford, MA, 1973.

[9] Wm. Arthur Conklin, Raymond E. Cline, Jr., and Tiffany Roosa, "Re-engineering Cybersecurity Education in the US: An Analysis of the Critical Factors", in *Proceedings of the 47th Hawaii International Conference on System Science*, 2014, pp. 2006-2014.

[10] ABET, "Engineering change: Lessons from leaders on modernizing higher education engineering curriculum", Accreditation Board for Engineering and Technology, Inc., Fall Issue Brief, 2017.

[11] Willis H. Ware, "Security controls for computer systems: Report of defense science board task force on computer security", Report R-609-1, RAND Corporation, 1970.

[12] Jerome Saltzer and Michael Schroeder, "The Protection of Information in Computer Systems", in *Proceedings of the IEEE*, 63(9), 1975, pp. 1278-1308.

[13] Peter J. Denning, Douglas E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, and Paul R. Young, "Computing as a Discipline", *Communications of the ACM*, 32(1), 1989, pp. 9-23.

[14] Gary A. Ford and Norman E. Gibbs, "A Master of Software Engineering Curriculum: Recommendations from the Software Engineering Institute", *IEEE Computer*, 22(9), 1989, pp. 59-71.

[15] Wolfgang A. Halang, "A curriculum for real-time computer and control systems engineering", *IEEE Transactions on Education*, 33(2), 1990, pp. 171-178.

[16] Cynthia E. Irvine, "Goals for Computer Security Education", in *Proceedings of the IEEE Symposium on Security and Privacy*, Los Alamitos, CA, 1996, pp. 24-25.

[17] Cynthia E. Irvine, Shiu-Kai Chin, and Deborah Frincke, "Integrating Security into the Curriculum", *IEEE Computer*, 31(12), 1998, pp. 26-30.

[18] John Biggs and Catherine Tang, "Teaching for Quality Learning at University, 4th ed.", Open University Press, Berkshire, UK, 2011.

[19] Bertrand Meyer, "Software Engineering in the Academy", *IEEE Computer*, 34(5), 2001, pp. 28-35.

[20] T.C. Lethbridge et al., "SE2004: Recommendations for Undergraduate Software Engineering Curricula", *IEEE Software*, 23(6), 2006, pp. 19-25.

[21] R. A. Clark and H. A. Schmidt, "A national strategy to secure cyberspace", The President's Critical Infrastructure Protection Board, Washington, DC, 2003.

[22] Samuel T. Redwine (ed.), "Software Assurance: A Guide to the Common Body of Knowledge to Produce, Acquire and Sustain Secure Software, Version 1.1", U.S. Department of Homeland Security, Washington, DC, 2006.

[23] Nancy R. Mead et al., "Software Assurance Curriculum Project Volume I: Master of Software Assurance Reference Curriculum", Software Engineering Institute, Carnegie Mellon University, Technical Report CMU/SEI-2010-TR-005, 2010.

[24] Carol Woody, Nancy Mead, and Dan Shoemaker, "Foundations for Software Assurance", in *Proceedings of the 45th Hawaii International Conference on System Sciences*, 2012, pp. 5368-5374.

[25] Tom Hilburn and Dan Shoemaker, "Engineering Competencies", in Nancy R. Mead & Carol C. Woody (eds.) *Cyber Security Engineering: A Practical Approach for Systems and Software Assurance*, SEI Series in Software Engineering, Addison-Wesley, 2017.

[26] Diana L. Burley, Matt Bishop, Scott Buck, Joseph J. Ekstrom, Lynn Futcher, David Gibson, Elizabeth K. Hawthorne, Siddharth Kaza, Yair Levy, Herbert Mattord, and Allen Parrish, "Cybersecurity Curricula 2017", Version 1.0 Report, December 2017.

[27] Jessica Anne Chisholm, "Analysis on the perceived usefulness of hands-on virtual labs in cybersecurity classes", Ph.D. Thesis in Computer Science, Colorado Technical University, 2015.

[28] Daniel Conte de Leon, Christopher E. Goes, Michael A. Haney, and Axel W. Krings, "ADLES: Specifying, deploying, and sharing hands-on cyber-exercises", *Computers & Security*, 74, 2018, pp. 12-40.

[29] DEFCON, "DEFCON 26 - Capture the Flag", Website. Available: https://defcon.org/html/defcon-26/dc-26-ctf.html

[30] ECSC, "European Cyber Security Challenge", Website. Available: https://www.europeancybersecuritychallenge.eu

[31] CCDC, "U.S. National Collegiate Cyber Defense Competition", Website. Available: http://nationalccdc.org

[32] Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin, "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains", in *Proceedings of the 6th International Conference on Information Warfare and Security*, 2011, pp. 113-125.

[33] Chloë Marshall, "Montessori Education: A Review of the Evidence Base", *Science of Learning*, 2(11), 2017.